SECURITY CHAMPION'S GUIDE TO
WEB APPLICATION SECURITY

What a web application firewall can do to defend your web applications

Akamai

Who Should Read This e-Book

Application security champions are the main audience for this e-Book. Application security champions come from many diverse backgrounds, but they are unified by a determination to address a longstanding, pernicious problem – web application vulnerabilities.

An application security champion can be a developer, an architect, a traditional information security role, or a technical manager. This guide is intended to help all types of application security champions ask for more from web application security and to develop and deploy more effective web application security capabilities.

"At the end of 2018, less than 20% of enterprises will rely only on [network] firewalls or intrusion prevention systems to protect their web applications – down from 40% today.

By year-end 2020, more than 50% of public web applications protected by a web application firewall (WAF) will use WAFs delivered as a cloud service or Internet-hosted virtual appliance – up from less than 10% today."

// GARTNER[1] "Magic Quadrant For Web Application Firewalls" by Jeremy D'Hoinne, ADAM Hills, Greg Young, July 15, 2015

How to Use This e-Book

This e-Book describes the key security capabilities that a web application firewall (WAF) can deliver to defend your web applications. Readers can use this information to:

- Consider whether a WAF is appropriate for your web application security
- Baseline your existing WAF against a set of recommended criteria
- Generate ideas for extending an existing WAF and improving web application security

Purpose

No single security tool can achieve all security goals. It is important for security architects to find a balanced, layered approach that raises the floor of what is minimally acceptable, and elevates the overall set of what is possible – the ceiling of what tasks, threats, and vulnerabilities the security architecture can manage.

Security is a tricky business. The discipline is filled with concepts such as least privilege that are easy to request but hard to implement in practice and at scale. Enterprise security teams must deliver security capabilities that integrate into a living, breathing software organization, that can cope with determined attackers, and that can scale to meet the needs of the enterprise in the future.

WebApp Sec 101

Security is a game of weakest links. Attackers need to find a weak link to gain access. Defenders cannot rely only on check boxes and spreadsheets to keep them safe. Instead we must identify and strengthen our weak links, further strengthen our strong links, and ensure the system can scale.

Attackers innovate: they adapt and add new capabilities to their toolkits. Defenders must stay up to date. Look for security technologies that scale to the large traffic volumes that web applications require, defenses that are not static and can adapt like attackers do, and security services that play nice by integrating well with applications and use cases.

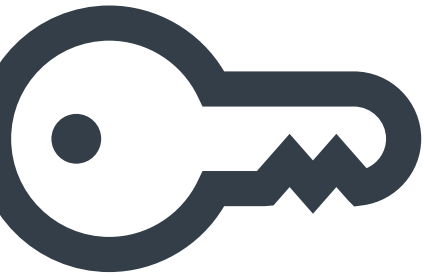
Web application security is of strategic importance to the enterprise. In the past, the challenge was to build awareness and get the executives and developers to care about security. That's no longer the case. Reading the daily headlines, everyone from the CIO to the CEO to the executive board is now actively investing time and resources into security. Now for the hard part – what do we do next to protect our users and the data on the web? That's what this e-Book is about.

The State of Security Professionals Today

Security pros operate a bit like chess masters: dealing with wily, adaptable opponents who are armed with robust attack methods.

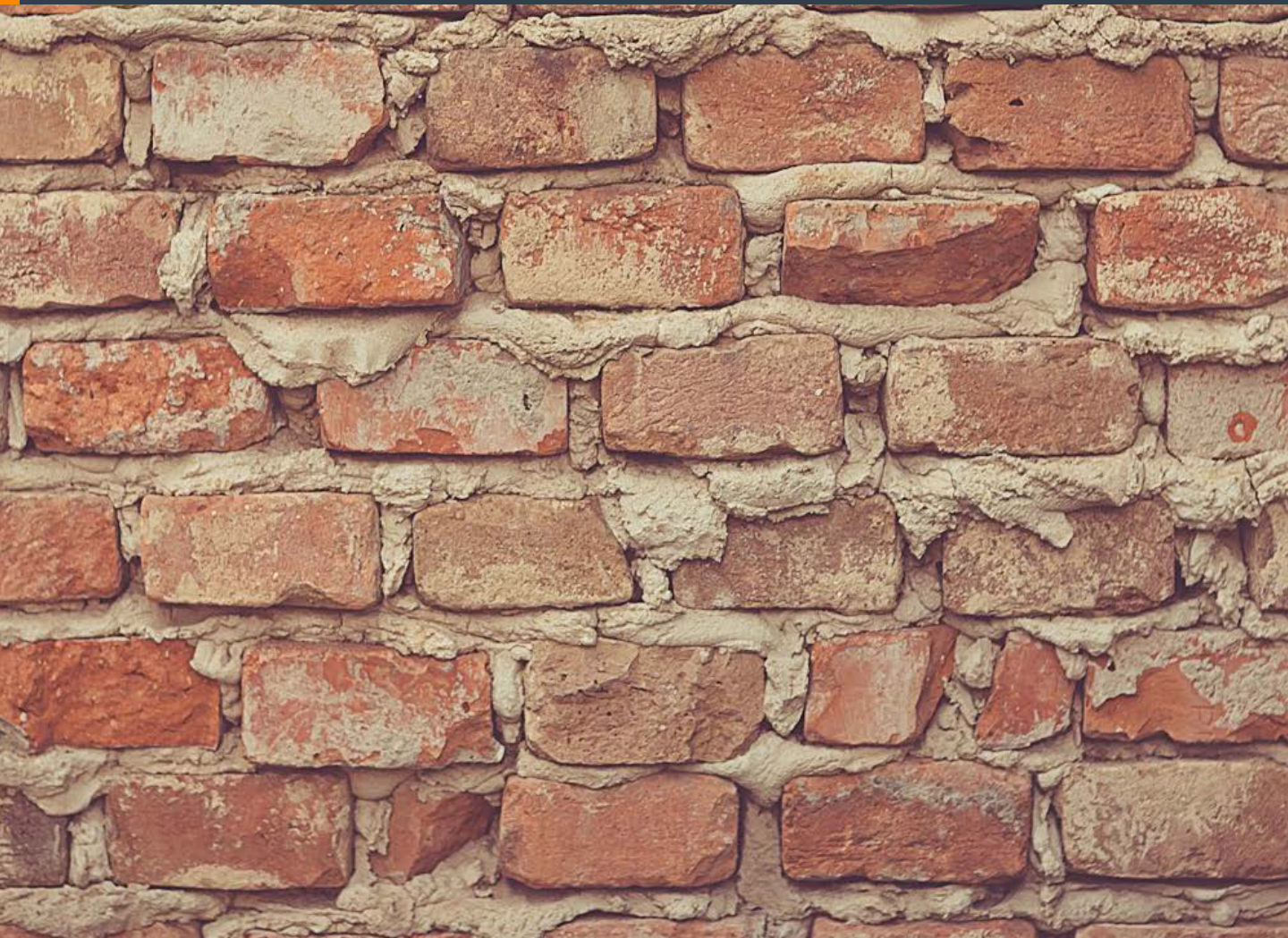
There is no one best move in chess, nor is there a best product in security architecture. Neither is there a perfect defensive formation. Instead it is about finding the best available move for your situation. In a perfect world, software would be written flawlessly, WAFs would not be needed, and attackers would be impotent. In the world we live in, these are not the case.

To find the best available move given your starting position is a four-step process: First, analyze your starting position. Second, consider the threats against your system. Third, limit the options of your opponent. Fourth, develop the strengths of your own system.



CHAPTER BREAKDOWN

CHAPTER 1 Behavioral Perimeter



The classic firewall,

barrier to block flames, is based on structure. Endpoints are either inside or outside the firewall. Unfortunately, application defense today is not an inside versus outside business. Attackers seek out system structures and design ways to go through, over, around, and under them. Structures remain important in security architecture, but they are insufficient to deal with the adaptable behavior of attackers.

Adaptability is a signature feature for skilled attackers, and the time is now for defenders to learn how to add behavioral adaptability to the defensive playbook. Security protocols and services should build in flexibility and depth to mediate a wider range of events and interfaces. In short, security is not delivered only to a topology, but rather to a chain of behavior-based events.

Stopping bad behavior requires a behavioral perimeter more so than a high or fireproof wall. Behavioral perimeters are less like walls and more like pipelines through which events flow. A security pipeline defines and enforces a specific set of checks, a specific order of operations, and visibility monitoring. Characteristics of a security pipeline include:

Type and nature of the security services in the pipeline

Ease of adding new services

Ability to customize security services

Visibility monitors

Protocol support

Self-protectedness - features that protect the pipeline itself

The old network firewall set out to establish a structural perimeter where the touchpoints for integration were either inside or outside the firewall. In contrast, a behavioral perimeter has many more touchpoints to integrate and extends both more broadly across applications and more deeply into application logic and data.

CHAPTER 2 Security at Scale

Information security has a broad set of goals – confidentiality, integrity, and availability. The last one – availability – really gets the attention of the whole business. Downtime translates directly into higher costs and lower revenues. Maintaining availability and minimizing latency is essential as you scale.

Scale gives visibility

A key principle of operating at a large scale comes from Amazon's James Hamilton[2] in, "At Scale, the Incredibly Rare is Commonplace." Hamilton refers to issues like disk failure, which are mainly rare issues for an individual device but become commonplace with thousands of disks, making disk failure a daily event. The same is true in security.

Metering – scale wisely

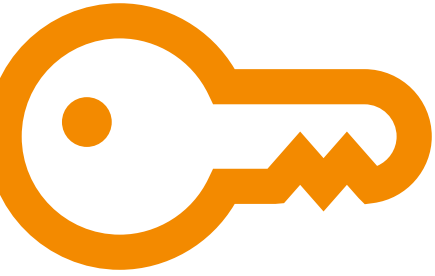
Scaling implicitly requires metering functionality to deliver resources effectively. Metering and monitoring are at the foundation of one of the most crucial areas of web application security – risk-based access control. For example, metering and monitoring can help identify brute force and lateral attacks. It's not enough to passively react to attackers with basic access control strategies, however, because they've already gamed and mapped out such systems. Access control must adapt and learn from daily activity by legitimate users and malicious ones alike.

Loose coupling, scale and margin of safety

The loosely-coupled architecture of the web is a threat and an opportunity for security architects. Threats come from unmanaged and unprotected data caches and lack of coordination of the overall set of security protocols.

Security opportunities abound in loose coupling, however. First and foremost is the ability to insert a specialized security layer: a WAF upstream of the web application. The WAF adds a margin of safety to inspect and block malicious content (such as SQL injection) before the content reaches the application that it is targeting. Security architecture is as much about what you choose to do, such as data sanitization and input validation, as where you choose to do it – inside the application, in a filter, or upstream of the application in a WAF. It's better to run the security checks outside the context of what the security mechanisms protect.

[2] http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_reInvent20121128.pdf



Latency & traffic spikes

Perhaps you have heard that people have short attention spans these days? Latency in your website will result in many users moving along to another site. Research from Amazon, Google, and others[3] shows that slow response results in lost dollars due to user abandonment of transactions.

Security protocols are notorious performance hogs, so it is necessary to ensure a WAF does not degrade response time beyond an acceptable point. Latency is a long-standing problem that factors into every security decision. The best security protocols for a business are the ones that minimize customer-facing latency.

To offer effective security, a WAF must be in line all the time, so it can block malicious traffic – not just watch as it passes. As such, a WAF has to handle peak loads and still not introduce excess latency. Users come and go in waves, and traffic bursts are a reality for successful sites, so a WAF must manage normal traffic and unusually high traffic. It must be able to handle large-scale traffic spikes that may be brought on by customer interest, seasonality, or a popular offer on the site.

Higher traffic volumes can mean more places for attackers to hide, so an inline WAF is just as important or more important on Cyber Monday. A retailer should not have a WAF that must be turned off or placed out of line because it can't handle seasonal traffic loads.

As a site scales, errors, security findings, false positives, and false negatives from a WAF must be kept to a minimum, so that peak traffic does not swamp the security staff with false findings.

[3] http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html

Security coverage

Your security protocols and access control can only be as good as the level of security coverage across your application portfolio. Every service in line with a critical web application must scale. As a result, your ability to scale is limited by the reach and coverage of your security services.

A key to access control is to ensure that the access control policy enforcement point cannot be bypassed. Attackers are highly skilled at reconnaissance, for example by crawling an entire site (spidering) to index all its visible contents. They specialize in finding side doors and back doors into applications such as admin screens, consoles, and other functionality. Attacker access can lead to a large compromise of a site and its data. As a result, a WAF must have the ability to cover all externally accessible functionality and to mediate communications. Ensuring that all communications for the web application's namespace are mediated via a gateway or proxies will result in a better night's sleep for the security champions.

Security as an enemy of scale

No such thing as a free lunch. Scaling, by itself, does not eliminate bottlenecks – it moves them to a place in the architecture where it is cost effective to add resources. Cookie, session tokens, and nonces (numbers used once) have valid roles in security design schemes, but they are also a challenge for scaling because they add state into the application architecture. State can inhibit functionality and must be planned for to ensure that it does not break anything. WAFs should be selected based on the breadth of the tools they bring to bear on state and session management.

IT'S LIKE FINANCIAL SCALE

A major factor in why businesses are pro-cloud is that it takes capital expense (CAPEX) off of their books and moves it to operational expense (OPEX). Sure, this sounds wonky to infosec pros (but, hey, imagine how we sound talking to the business about asymmetric and symmetric crypto operations). This accounting difference is a lesser-known reason why businesses are aggressively moving into the cloud. The move from CAPEX to OPEX is financial scale. Just as IT will not pull back from delivering more computing scale at lower cost, the business side will not reduce their efforts to lower capital expenses.

CHAPTER 3 Intelligent Security

Attackers do not limit themselves to old-school vulnerability scanners like SATAN and Nmap, so why is web application security often built on top of yesteryear's passwords and role-based access control? Sure, the old-school tools have a role to play (just like Nmap does), but they should be the starting point for building more advanced defenses – establishing the floor, not the ceiling.

Web protocols are mainly built on the HTTP request-response pattern. Even if a defender has built effective access controls, the attacker can use tools like fuzzers that throw random input at applications. Fuzzers discover coding errors and security loopholes by generating many millions of request permutations and combinations that yield a very large volume of HTTP responses. By studying these responses, attackers identify weak points in a system's architecture.

A major security weakness is a web system that does not adapt its responses based on the input. Instead of generating a static response, defenders can use learning systems to adapt a defense based on the input threat. Defenders need to co-evolve with their attackers' methods. It is long overdue.

SQL injection defense

SQL injection (SQLi) is a pressing problem for web applications. Margin of safety dictates that no single defense layer should be counted on to protect the most valuable resources. Defenders have a range of options that can be brought together into a defensive playbook to stop SQL injection in a multi-layer defense:

WAF – Content and behavior analysis, input validation, data sanitation

Application – Input validation, data sanitization

Back-end – Stored procedures, prepared statements

Since data is the attackers' ultimate goal, no one layer should be counted on to deliver 100% of the protection. Security is a chain of responsibilities. A holistic approach takes advantage of the capabilities of each defensive layer.

Defending against SQL injection can include inspecting input. If this route is taken, the inspection should ensure that the input is validated against allowable criteria to ensure it does not contain hostile code. Further protections should include escaping and sanitizing the data to normalize or strip out malicious characters.

In theory, programmers should ensure that their applications are not vulnerable to SQL injection. In practice, however, it takes only one programming error to allow a major data breach. A WAF is a very attractive option for adding a margin of safety to protect your users and their data. Network firewalls by themselves cannot do it, because they lack the application and data depth.

Cross-site scripting defense

Cross-site scripting (XSS) is among the most common web application attacks. The security industry spends a lot of time and resources dealing with zero-day attacks, but in the same way zero days can lead to high-impact attacks against enterprise software, user data is vulnerable to XSS. These vulnerabilities result in the release of credentials, passwords, tokens, and session cookies that are used for account takeover, the compromise of user information, and other high-severity damage to users.

The WAF layer is an ideal location to perform XSS countermeasures. Output encoding can be used to separate the data plane from the control (executable) plane. This separation is usually performed as close to the outbound communication (origin) as possible. The WAF layer can use its last line of defense to ensure that content is encoded properly at the edge.

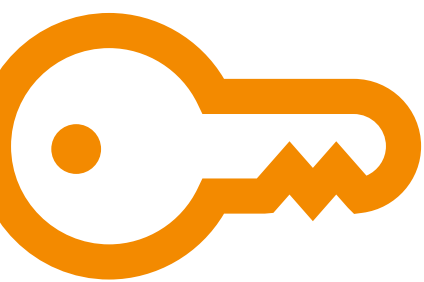
Output encoding should include all data sent back to clients, such as HTML, URLs, JavaScript, and XML. A WAF can deliver additional XSS defenses by ensuring the correct implementation of browser directives such as cookie flags and content security policies such as CSP[4].

Cross-site request forgery defense

A robust response to cross-site request forgery (CSRF) attacks may involve a range of tactics. The most effective option is to use a cryptographic nonce to add per-user, per-click or per-transaction state tracking to each request. WAFs are not the ideal layer at which to conduct this check, because defending against CSRF with cryptographic nonces is best done at the intersection of user, application, and session state. A WAF will lack some important context at its traditional location at the edge of the system.

Nonetheless, security systems are built up in layers. Just because a WAF is not the ideal layer in which to deliver a knock-out punch to a CSRF does not mean it does not have a role to play. Although not a complete CSRF defense, a WAF can check the HTTP Referrer and Origin to verify the origination of the request and block possible CSRF attacks.

[4] http://www.w3.org/TR/2014/WD-CJP11-20140211/



Remote file inclusion defense

Successful remote file inclusion (RFI) attacks give the attacker the ability to send executable commands, files, and scripts over the data plane and run them on the web server. The WAF layer is in a good position to implement many core RFI defenses, because it is at the boundary between external and internal systems. Core defense techniques include checking input data for malicious executables, explicitly sanitizing input data, and implementing an indirect object reference map.

The goal of checking and sanitizing input data for RFI is to identify and block data that contains malicious code. An indirect object reference map ensures the back-end resource name is not used on the user-accessible front end. It works like a hash or index to limit an attacker's ability to directly manipulate back-end resources.

CHAPTER 4 Integration

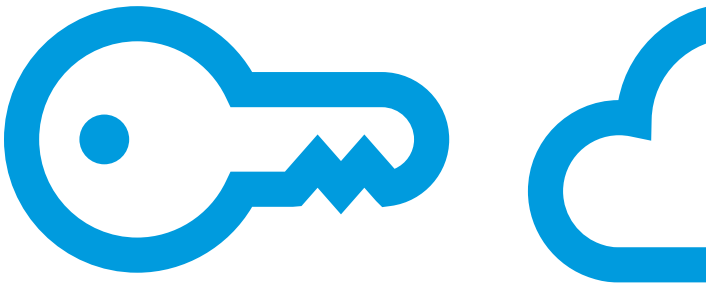
Many good security designs do not deliver good security, not because the defensive theory is unsound, but because the designers cannot achieve integration into real-world systems. Consider public key infrastructure (PKI), which proposed to deploy a truckload of capabilities including integrity, strong authentication and encryption. Why did PKI not take root in the architectures of the dotcom era when PKI was born? Because the products lacked integration into clients, servers, and applications.

Engineering is all about learning from mistakes. One lesson from PKI is to not underestimate the importance of integration. Do not get wrapped in what a security service can do, instead stay focused on which of its qualities your system can reasonably absorb.

When it comes to integration, WAFs have an advantage over many other types of security services – WAFs are designed to be deployed upstream of the application. As a result, they form a natural boundary around the application.

Boundary design

Defining an effective boundary requires thinking through the integration layers. There are two classes of integration to factor into the design: technical and process. Aspects and questions related to each type of integration are shown in Figure 1.



TECHNICAL INTEGRATION	
Front-end integration	To clients, browser, mobile apps, and sessions
Back-end integration	To web applications, APIs, web services, and messaging systems
Session integration	Does the WAF authenticate? How does it deal (if at all) with session and page state?
Deployment integration	Are the WAF and the web application deployed on premises, or in the cloud, or both?
PROCESS INTEGRATION	
Software development lifecycle (SDLC) integration	What tools, frameworks, SDKs, and clients do the developers need to build code for a WAF-enabled web application?
Tester integration	What tools are required to do functional tests?
Security testing integration	How to conduct ongoing vulnerability assessment and penetration testing for WAF-enabled web applications
Risk services integration	How to test and monitor efficacy of risk-based control and feedback loops

Figure 1: The two types of integration required for boundary design

Integration is the missing link in most security architectures and a leading cause of failed projects and failed deployments, not because it is impossibly difficult, but because it is generally an afterthought – as if it were simply an implementation detail. Instead, integration should be a component of the security architecture. Network security services are like walls, but less hard. The security architect should have a from-to mindset that funnels front-end access to back-end resources and applies security policies. It is easy to focus on the security policies alone and miss the communication channels, adapters, filters, and messages that enable integration.

CHAPTER 5 DevOps Meets SecDevOps

Long-term success is determined as much by taking best advantage of people's skills as it is by any architecture. Operational staff is moving aggressively towards a DevOps model, which uses programming models and tools to help scale operations. The same basic approach found in DevOps is needed in security.

What could SecDevOps look like? While there is not a cohesive movement today, there are some interesting examples in the field. One promising example is embodied in Gauntlt, a tool in use at organizations that operate at very large scale.

Gauntlt brings together security tools such as SQLMap, cURL, SSLyze, Nmap, DirBuster, and others, and inserts them into the Agile SDLC, allowing these security tests to run alongside standard functional and unit tests. Security failures can be found far earlier in the development process, especially when the tests run from the very first build instead of at a penetration testing (pentest) two weeks before go live.

This change means that the security team operates more like a traditional development organization. They are not siloed. Security organizations add value to the SDLC by bringing improved AppSec designs and capabilities like WAFs and by bringing test cases to help find and avoid mistakes.

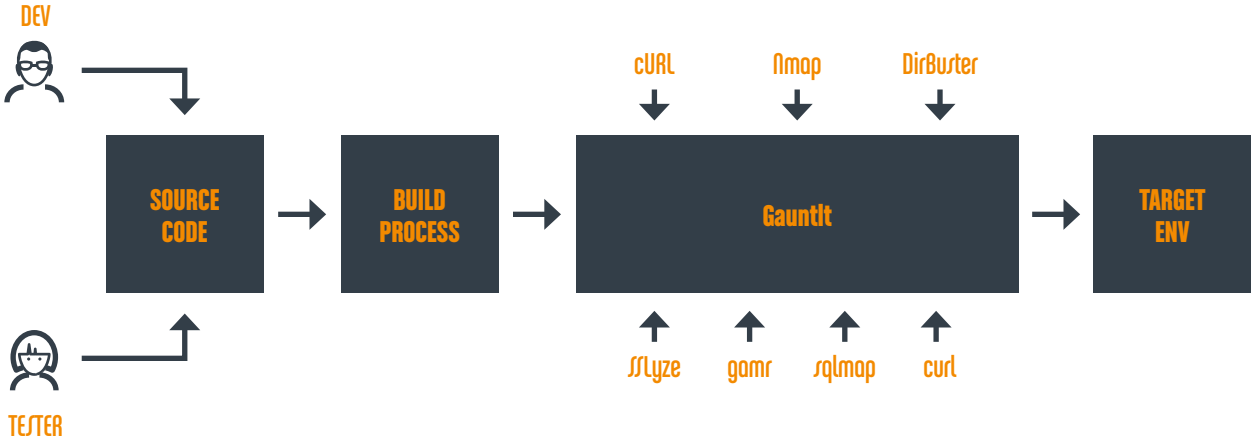


Figure 2: Running the Gauntlt. Gauntlt analyzes code for vulnerabilities before promoting the code to production

Effective security test instrumentation

Good security requires an effective and continuous security testing environment. There is no single out-of-the-box solution. Some effort is required to build and train an attack dog that you can unleash on your app. The results can pay dividends, because automated tests scale better than any manual pentest and, properly tuned, they find vulnerabilities missed by untuned commercial scanners. As with most aspects of security, an end-to-end view is very helpful. The tester should aim for:

- Coverage across the entire accessible site. Avoid telescoping tests that only review a subset.
- Wire-on tests that log in and follow logic
- Simulate malice, such as stolen credentials, cookies, and session tokens
- Test security features, too. A study from Veracode[5] showed that security products themselves have the highest vulnerability density. Test access controls and other security technologies even more rigorously than application functionality.

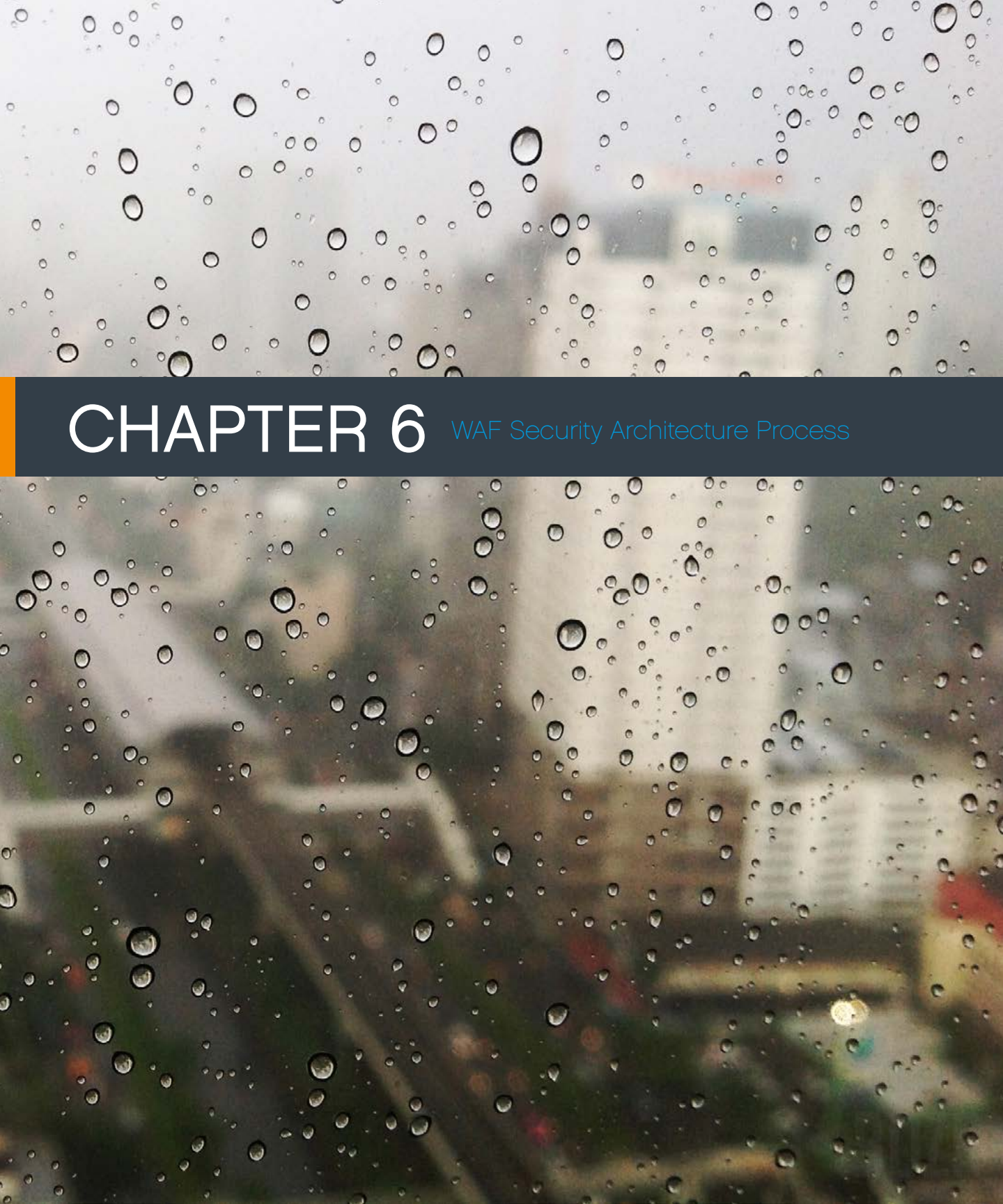
Ruleset tuning: dealing with roses and thorns

New technologies and new threats are a fact of life in security. One of the benefits of security tests is that they provide clear input to the ruleset-tuning process. As new attacks appear, they can be run through the security testing process. The security team can monitor the effectiveness of the existing rulesets at blocking the new attacks and determine if they need to be updated.

The same approach works with new technologies. If an application is ported from WebSphere to Ruby on Rails, it is critical to determine if the rulesets still block XSS, SQLi injection, and other relevant attacks.

Rulesets should be carefully reviewed, tested and updated on an ongoing basis, not just once a year when the auditor comes in. Whether it is outsourced or the responsibility of someone in your organization, the WAF should have a clear owner who is responsible for keeping the rulesets fresh and on point with the current attack landscape.

[5] http://www.veracode.com/veracode-study-reveals-internet-things-poses-cybersecurity-risk



CHAPTER 6 WAF Security Architecture Process

Security architecture isn't a fixed state; it is an ongoing process. Security services must adapt to the threat environment – reading the threat landscape, limiting attackers' options, and developing defender coverage.

The four primary iterations in WAF security architecture follow this sequence:

1. Understand the application
   a. Ensure coverage and scale
   b. Cannot be bypassed, always on
2. Analyze threats
   a. Behavioral, not static defenses
   b. Ongoing refinement, always in learning mode
3. Limit attack space
   a. Minimize attack surface
   b. Shrink room for attacker to operate
   c. Compensating transactions
   d. Monitoring
   e. Intelligent, adapts
4. Develop
   a. Extend application capabilities in harmony with security protocols and defensive services
5. Iterate - Return to 1

The set of security capabilities becomes the organization's living, breathing boundary. A web application security playbook describes how to drive the capabilities through design, development, testing and deployment.

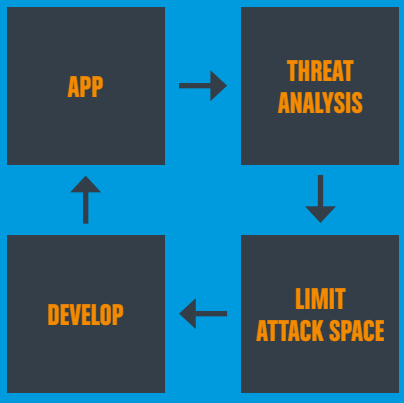
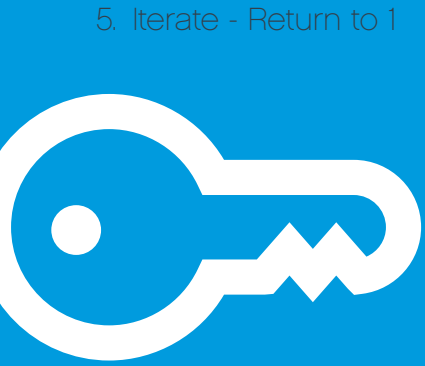
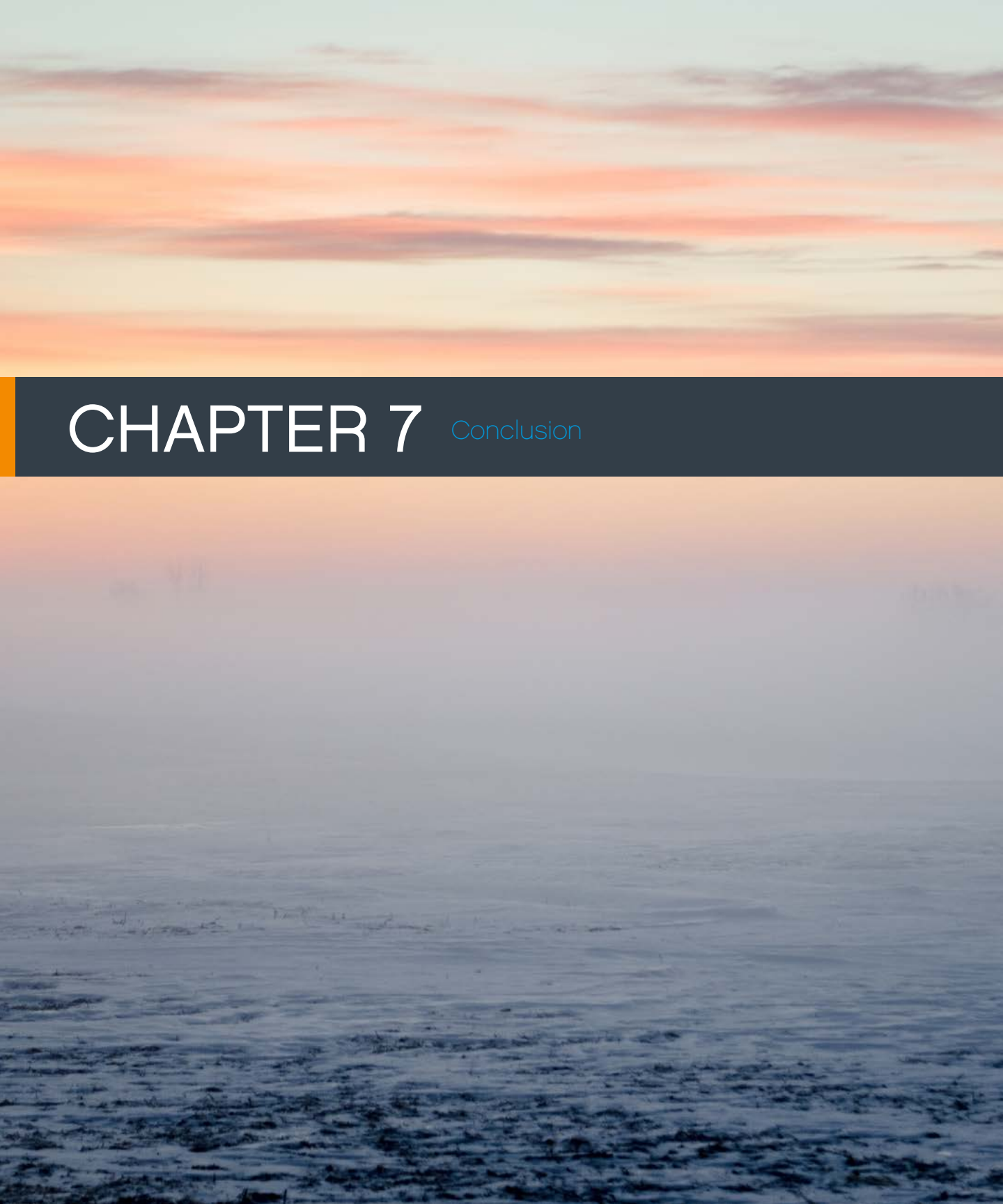


Figure 3: Security architecture is an ongoing process





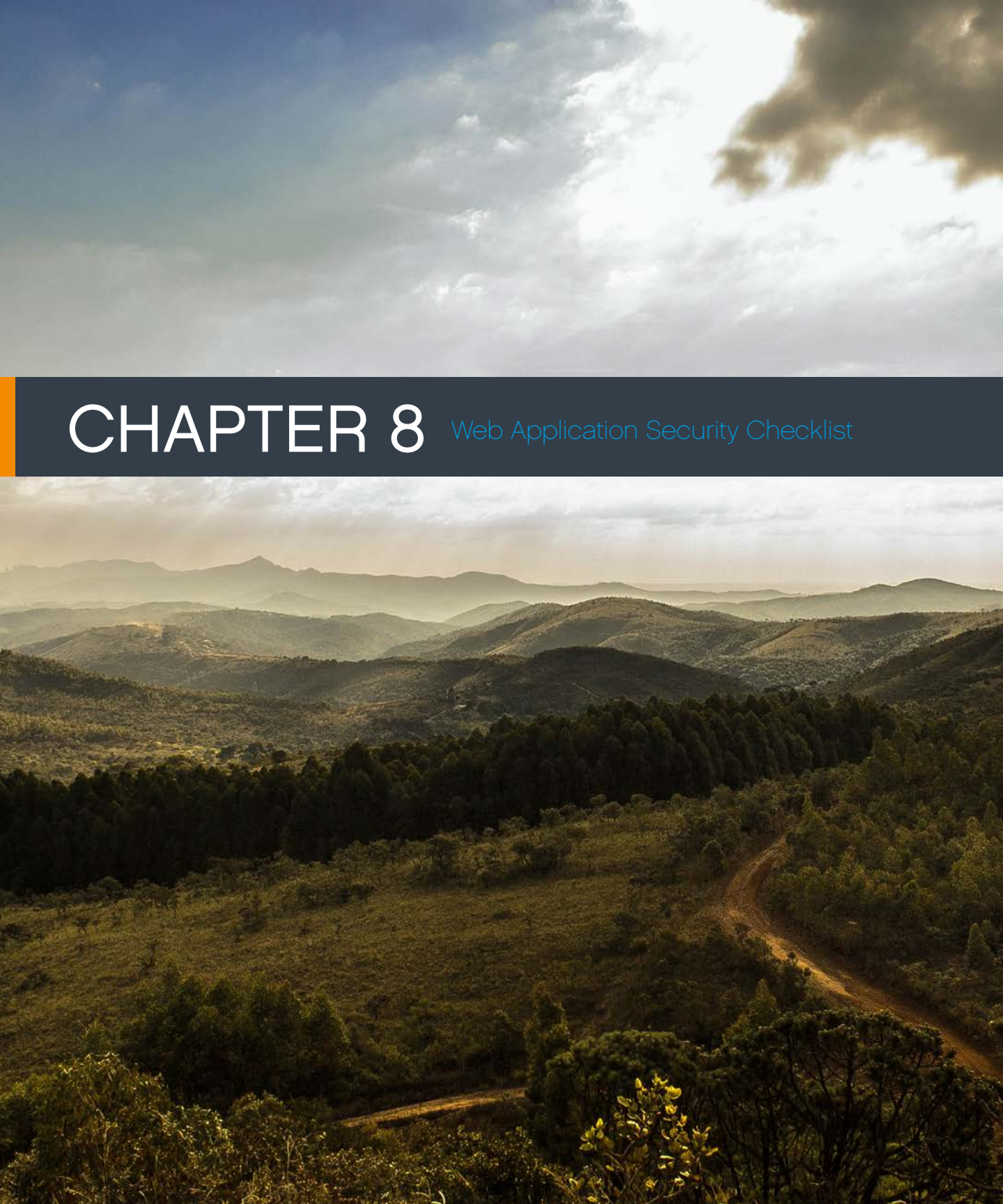
CHAPTER 7 Conclusion

Security is a chain of responsibilities.

The job of security architecture, design, and operations requires that you have a clear idea of what capabilities are performed and where they are executed. Due to its placement on the edge of the system, the WAF layer is among the most critical web security elements.

The single characteristic that separates the security mindset from the rest of web application development is that the goal of developers and architects is to create applications that work, while the goal of security architects is to plan and design for failure modes. From an assurance standpoint, security architects cannot rely on a single layer of defense to deliver an adequate margin of safety against determined and innovative attackers. The system as a whole is far more reliable and defensible with a WAF layer upstream to offload and optimize specific security services.

A WAF layer can potentially offer improvements in important areas – the ability to handle large volumes, visibility and intelligence, and simplicity through centralizing security logic. These benefits are a critical part of a balanced approach to web security.



CHAPTER 8 Web Application Security Checklist

This section summarizes key action points for WAF security architects to include in a web application playbook.

ARCHITECTURE

- Understand the application
  - » Functional use cases and business context
- Coverage and scale requirements
- Develop security architecture framework – logical, development, assurance, and operational views

DESIGN

- Analyze threats
- Define design to:
  - » Raise the floor – minimally acceptable
  - » Raise the ceiling – fundamentally new capabilities
- Limit attack space
  - » Monitoring
  - » Intelligence, adaptation

DEVELOPMENT

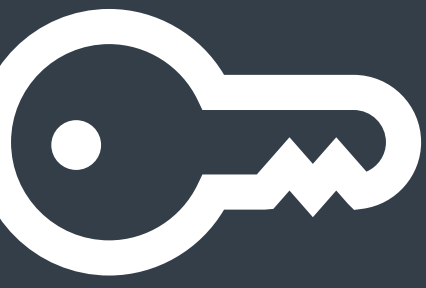
- Extend application capabilities in harmony with security protocols and defensive services
- First mile integration – to client communications
- Last mile integration – to back end
- SecDevOps test case instrumentation

TESTING

- Unit tests
- Synthetic testing – end to end
- Vulnerability assessment
- Penetration testing

DEPLOYMENT AND OPERATIONS

- Operational monitoring
- Metering
- Usage data feedback to improve visibility into user activity and transactions



VISIT OUR BLOG http://blogs.akamai.com

CALL 1.877.425.2624

If you'd like to speak with a representative.

CLICK Email Sales

To contact our sales team today

As the global leader in Content Delivery Network (CDN) services, Akamai makes the Internet fast, reliable and secure for its customers. The company's advanced web performance, mobile performance, cloud security and media delivery solutions are revolutionizing how businesses optimize consumer, enterprise and entertainment experiences for any device, anywhere. To learn how Akamai solutions and its team of Internet experts are helping businesses move faster forward, please visit www.akamai.com or blogs.akamai.com, and follow @Akamai on Twitter.

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 57 offices around the world. Our services and renowned customer care enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers and contact information for all locations are listed on www.akamai.com/locations.

Akamai